제스처 및 음성 인식을 이용한 윈도우 시스템 제어에 관한 연구

김 주 홍*, 진 성 일*, 이 남 호**, 이 용 범**
*경북대학교 전자공학과 **한국 원자력 연구소

Study about Windows System Control Using Gesture and Speech Recognition

Ju-Hong Kim, Sung-Il Chien , Nam-Ho Lee and Yong-Bum Lee
*Department of Electronics Kyungpook National University
** Korea Atomic Energy Research Institute
email: kimjh@palgong.kyungpook.ac.kr

Abstract

HCI(human computer interface) technologies have been often implemented using mouse, keyboard and joystick. Because mouse and keyboard are used only in limited situation, More natural HCI methods such as speech based method and gesture based method recently attract wide attention. In this paper, we present multi-modal input system to control Windows system for practical use of multi-media computer. Our multi-modal input system consists of three parts. First one is virtual-hand mouse part. This part is to replace mouse control with a set of gestures. Second one is Windows control system using speech recognition. Third one is Windows control system using gesture recognition. We introduce neural network and HMM methods to recognize speeches and gestures. The results of three parts interface directly to CPU and through Windows .

I. 서 론

최근 사람과 컴퓨터 사이의 인터페이스 기술은 키보드, 마우스, 조이스틱 등과 같은 장비를 이용하는 것에서부터 음성 및 제스처등을 기반으로 한 실제로 사람들이 대화하는 것과 같은 유사한 기술의 개발에 중점을 두고 있다. Fels 등은 Glove-Talk, Glove-Talk II[2] 에서 핸드 제스처를 인식하여 음성으로 합성하였고, 한국 과학 기술원의 변 증남 교수팀은 수화동작을 인식하여 그래픽 및 텍스트로 출력하는 시스템을 완성하였다. 한 영원 등은 Windows 95 환경에서 음성 인터페이스 시스템[5]을 구현하였다. 본 논문에서는 음성과 제스처를 입력받아 윈도우를 제어[6][7]할 수 있는 다중 모드 입력 시스템을 구현하였다. 정적인 동작을 얻기 위한 센서 글러브로 사이버 글러브(CyberGlove)를 사용하였고, 동적인 동작을 얻기 위한 위치 센서로 폴리머스(Polhemus) 센서를 사용하였다[1]. 또한 마이크로폰을 통해 음성 데이터를 입력받았다. 폴리머스 센서를 통해서 얻은 3차원 위치 정보 및 사이버 글러브로부터 얻은 손동작을 인식하여 윈도우 시스템의 마우스를 제어하고, 연속된 동적인 동작에서 방향성 벡터를 추출하여 키보드를 제어하고 응용 프로그램을 실행시킨다. 또한 입력된 음성으로부터 실제 음성의 시작점과 끝점을 추출한 후 PLP 계수[4]를 얻어 음성을 인식하여 키보드 제어 및 응용 프로그램을 실행한다. 제스처 및 음성을 인식하기 위하여 신경회로망과 HMM[3]을 사용하였다. 인식된 음성 및 제스처 결과는 윈도우의 마우스 이벤트, 키보드 이벤트를 발생하여 제어하고 응용 프로그램을 실행시킨다. 이와 같은 다중 모드 입력 시스템은 이동하거나 혹은 컴퓨터와 떨어져있는 상황등에서 입력이 가능하여 사용자들이 좀더 자유로운 환경하에서 컴퓨터를 이용할 수 있게 하였다.

II. 시스템 구성

마이크로폰을 통하여 입력된 음성 신호는 16kHz 샘플링 주파수에 샘플 당 16비트로 A/D 변환된다. 제스처 데이터를 얻기 위하여 사이버 글러브와 폴리머스 센서를 사용한다. 사이버 글러브는 광섬유로 구성되어 있으며 손가락 마디 사이의 각도 및 손가락 사이의 벌어짐 정도 등을 18개의 센서로부터 얻을 수 있다. 사이버 글러브 시스템은 오른손에 착용할 수 있는 사이버 글러브와 사이버 글러브 인터페이스 유니트(CGIU)로 구성되어 있다. 그리고, 폴리머스 센서는 자

장을 발생시키는 트랜스 미터와 자장을 감시하는 리시버(receiver)로 구성되어 있으며 트랜스 미터와 리시버가 직렬 케이블로 SEU(stand-alone unit)에 연결되어 있다. 폴리머스 센서로부터 x, y, z 3차원 좌표 데이터를 얻을수 있다.

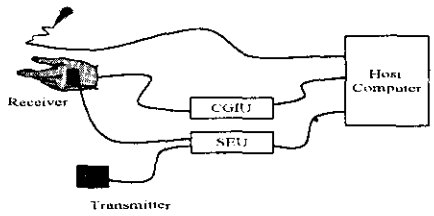


그림 1. 윈도우 제어를 위한 시스템 구성
Fig.1. Components of windows control system

III. 윈도우 제어 시스템

본 논문에서는 가상 핸드 마우스(Virtual-hand mouse)를 생성하여 윈도우의 마우스 이벤트를 생성시켜 마우스를 제어한다. 그리고 음성 및 제스처를 통해서 윈도우의 키보드 이벤트를 생성하여 키보드를 제어하고 응용 프로그램을 실행시킨다.

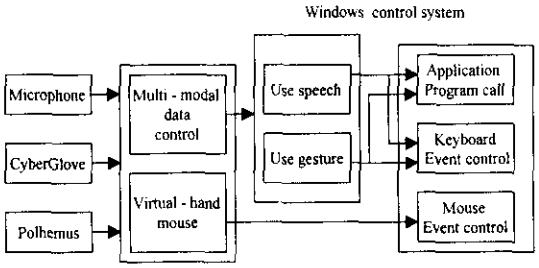


그림 2. 윈도우 제어 시스템
Fig. 2. Windows control system

가상 핸드 마우스는 폴리머스 센서의 위치 정보를 이용하여 모니터상의 커서 위치를 제어하고 사이버 글러브로부터 얻은 손모양 정보를 이용하여 버튼의 클릭 상태를 제어한다. 또한 손모양 정보를 이용하여 다중 모드 입력 시스템의 음성 및 제스처를 이용한 윈도우 제어 시스템을 가동시킨다. 가상 핸드 마우스의 위치를 제어하기 위하여 폴리머스 센서에서 얻은 3차원 좌표로부터 방향성 벡터를 추출한다. 방향성 벡터(V_t)는 샘플링 시간이 t 일때 좌표와 t-1일 때 좌표의 차로서 단위 시간당 x, y, z 축의 변화량을 가지게 된다.

$$V_t = (\Delta x_t, \Delta y_t, \Delta z_t) \quad (1)$$
$$where\ \Delta x_t = x_t - x_{t-1}, \Delta y_t = y_t - y_{t-1}, \Delta z_t = z_t - z_{t-1}$$

마우스는 2차원 정보만 필요로 함으로 z축 정보를 제거한 2차원 정보를 사용해 마우스 위치정보를 획득한다.

$$V_t(reduced) = (\Delta x_t, \Delta y_t) \quad (2)$$

가상 핸드 마우스의 위치는 이전 마우스의 위치와 마우스 위치의 변화량 V_t(reduced)의 합에 의해서 결정되어 진다.

$$Position = (x_p, y_p) + V_t(reduced) = (x_p + \Delta x_t, y_p + \Delta y_t) \quad (3)$$
x_p : previous x position, y_p : previous y position

사이버 글러브를 통해 입력받은 정적 동작을 이용하여 가상 핸드 마우스의 클릭 정보를 인식한다. 연속된 정적 동작들을 인식하여 왼쪽 버튼 클릭, 왼쪽 버튼 더블 클릭, 오른쪽 버튼 클릭, 드래그 등의 명령을 인식하여 마우스 이벤트를 발생시켜 I/O 관리자에 전달한다[7]. 인식기로는 신경회로망을 사용하였다. 동적인 클릭 동작을 몇 개의 기준 되는 정적 동작들의 연속으로 정의하고 연속적으로 입력되는 정적 동작을 인식하여 클릭 동작을 인식한다.

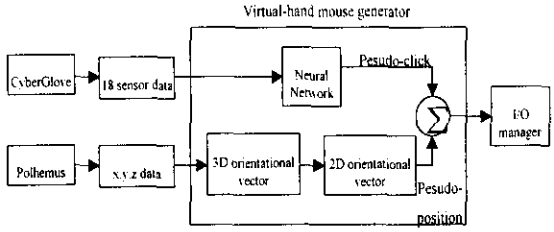


그림 3. 가상 핸드 마우스의 동작 원리
Fig. 3. Virtual-hand mouse operation principle

음성을 이용하여 키보드 제어 및 응용 프로그램 실행을 위해서 먼저 음성 데이터를 획득하고 특징 파라미터를 구하여 인식한 후 제어 명령을 발생시킨다. 음성 데이터를 처리하기 전에 잡음 부분을 제거해 주어서 실제 음성의 시작점(starting point)과 끝점(ending point)을 찾아야 한다. 음성의 시작점과 끝점을 찾기 위하여 음성신호의 에너지와 영 교차율(ZCR : zero crossing rate)을 이용하였다. 입력된 음성의 에너지는 프레임 단위로 구해지며 프레임의 일부를 겹치는 오버랩(overlapped) 방법을 이용하여 구하였다. 본 실험에서는 에너지와 비슷한 형태를 가지면서 계산량이 적은 평균 크기(M_n)를 구하여 사용하였다.

$$M_n = \sum_{m=-\infty}^{\infty} |x[m]w[n-m]| \quad (4)$$
$$w[n] = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & elsewhere \end{cases}$$
$x[m]$: *speech signal* , N : *integer*

식 4 에서 에너지를 구하고 일정한 값을 문턱값으로 설정하여서 시작점과 끝점을 추출한다. 하지만, 에너지를 이용한 시작점과 끝점 검출은 무성음 부분의 에너지가 작기 때문에 무성음 부분의 특성을 많이 잃어버릴 수 있으므로 영 교차율을 이용하여 2차 검출을 수행한다. 영 교차율은

$$Q[x[m]] = \sum_{m=-\infty}^{\infty} 0.5|sgn[x[m]] - sgn[x[m-1]]|w[n-m] \quad (5)$$

$$sgn[x] = \begin{cases} 1 & x>0 \\ -1 & x<0 \end{cases} \qquad w[n] = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & elsewhere \end{cases}$$

$x[m]$: *speech signal*

로 표현된다. 에너지를 이용하여 1차적으로 유성음 부분을 검출하고 영 교차율을 이용하여 최종적으로 무성음 부분을 검출하여서 음성 신호 중 잡음 부분을 제거한 실제 음성부분을 분리한다. 실제 음성 부분이 분리되면 PLP(perceptual linear prediction) 계수를 추출하고 신경회로망 인식기를 통과시켜 키보드 제어 명령 및 응용 프로그램 제어 명령들을 생성한다. PLP 모델[4]은 전극점 모델의 스펙트럼에 의해 근사된다. 귀가 느끼는 스펙트럼은 0-5kHz의 주파수 범위에서 16개 대역들로 나뉘어서 합하여지며, 중간대역과 상위대역을 보강하기 위하여 equal loudness pre-emphasis 를 거치게 된다. 또한 음성 스펙트럼의 전력 변화율을 감소시키기 위하여 3 제곱근 크기 압축을 실시한다. 이러한 처리를 거친 16개의 스펙트럼 성분에 푸리에 역변환 과정을 적용시켜 자기 상관 계수를 얻는다. 전극점 모델은 얻어진 자기상관 계수로부터 원하는 차수로 계산되며, 이로부터 다시 켑스트럼 계수를 계산할 수 있다. 본 논문에서는 5차의 전극점 모델을 사용하였다.

제스처를 이용한 윈도우 제어 시스템이 연속적으로 입력되는 동적인 동작 중에서 동작 명령 즉 윈도우 제어명령을 분리하기 위하여 정적인 동작을 이용하였다.

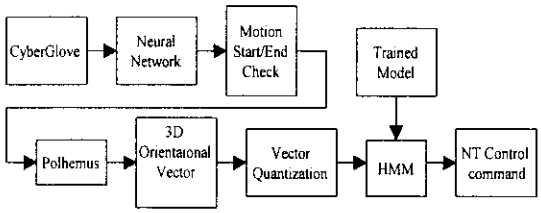


그림 4. 제스처를 이용한 윈도우 제어 시스템
Fig. 4. Windows control system using gesture

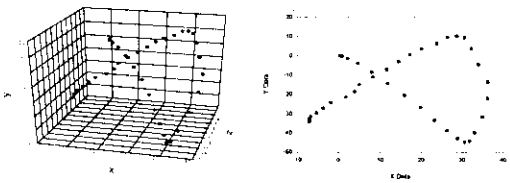
하나의 의미있는 동적인 동작은 식 6과 같이 시간에 따른 연속적인 방향성 벡터로 나타난다.

$$M = [V_1 V_2 \ldots V_t \ldots V_m]^t = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ \vdots & \vdots & \vdots \\ \Delta x_t & \Delta y_t & \Delta z_t \\ \vdots & \vdots & \vdots \\ \Delta x_m & \Delta y_m & \Delta z_m \end{bmatrix} \quad (6)$$

윈도우를 제어하기 위해서 기본단어 10개를 구성하였다. 구성된 단어는 의미있는 동적인 동작으로 구성된 단어이다. 이와 같은 동적인 동작은 수행하는 사람에 따라서 그 실행시간의 차이가 생겨 일반적인 신경회로망을 사용하여 인식하기가 쉽지 않다. 본 논문에서는 그러한 문제점을 해결하기 위하여 연속된 방향성 벡터들을 LBG 알고리즘을 이용하여 양자화하고 이산 HMM을 이용하여 인식하였다.

IV. 실험 결과

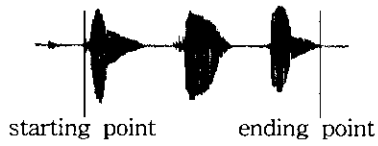
가상 핸드 마우스의 위치 정보는 폴리머스 센서로부터 얻은 3D 방향성 벡터를 2D 방향성 벡터로 변환하여서 구하고 마우스의 클릭여부는 사이버 글러브로부터 얻은 정적인 동작을 인식하여서 획득한다. 그림 5는 3D 방향성 벡터에서 변환된 2D 방향성 벡터를 이용하여 얻은 마우스 위치정보이다.



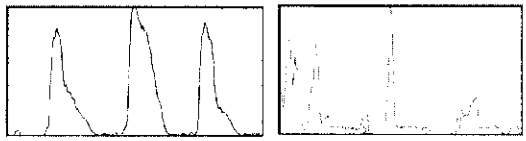
(a) (b)

그림. 5. 3차원 및 2차원 가상 핸드 마우스 위치 좌표
(a) 3차원 공간에서 위치 (b) 가상 핸드 마우스 위치
Fig. 5. 3D and 2D virtual-hand mouse position coordinate
(a) 3D space position (b)Virtual-hand mouse position

음성을 이용하여 윈도우를 제어하기 위하여 먼저 입력된 음성의 시작점과 끝점을 추출한 후 PLP 계수를 구하고 인식한 결과를 이용하여 키보드 및 응용프로그램 제어 명령들을 생성하여 윈도우를 제어하였다. 그림 6은 시작점과 끝점이 추출된 음성 파형을 보여준다.



(a)



(b) (c)

그림 6. 음성 신호, 에너지 및 영 교차율
(a) 시작점 및 끝점이 추출된 음성 신호
(b) 음성 신호의 에너지 (c) 음성 신호의 영 교차율
Fig. 6. Speech signal, energy, ZCR (zero crossing rate)
(a) Speech signal detected starting and ending point
(b) Energy of speech signal
(c) ZCR of speech signal

제스처를 이용하여 키보드 제어 및 응용 프로그램을 실행하기 위해서 폴리머스 센서에서 입력받은 3차원 좌표를 3D 방향성 벡터로 변환하여 양자화 한 후 HMM을 통해 제스처 명령을 인식한다.

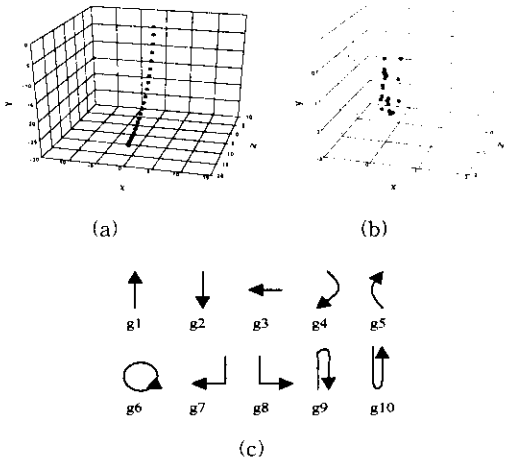


그림 7. 3 차원 좌표, 3D 방향성 벡터 및 제스처 명령
(a) 3차원 공간에서 동작 (b) 3D 방향성벡터
(c) 윈도우 제어를 위한 제스처 명령
Fig. 7. 3D coordinate and 3D orientational vector
(a) motion of 3D space (b) 3D orientational vector
(c) Gesture command for windows control

표 1. 응용 프로그램 호출 명령
Table 1. Application program calling command

음성 명령어	제스처 명령어	실행 프로그램
통신	g1	데이터맨 프로 실행
탐색기	g2	윈도우 탐색기를 실행
워드	g3	MS Word를 실행
인터넷	g4	MS Explorer를 실행
에디터	g5	Notepad를 실행

표 2. 키보드 제어 명령
Table 2. Keyboard control command

음성 명령어	제스처 명령어	제어 키	음성 명령어	제스처 명령어	제어 키
탭	g6	TAB	나누기	.	/
실행	g7	ENTER	영	.	0
탈출	g8	ESC	일	.	1
종료	g9	ALT-F4	이	.	2
왼쪽	.	←	삼	.	3
오른쪽	.	→	사	.	4
위쪽	.	↑	오	.	5
아래쪽	.	↓	육	.	6
더하기	.	+	칠	.	7
빼기	.	-	팔	.	8
곱하기	.	*	구	.	9

제스처 명령 g10은 윈도우의 시작 메뉴(Ctrl-Esc)를 실행한다.

V. 결 론

본 논문에서는 키보드, 마우스와 같은 제한된 입력 방법을 능가하는 다중 모드 입력 시스템을 제안하였다. 제스처를 이용해서 생성된 가상 핸드 마우스는 마우스의 위치 및 클릭 상태등을 제어한다. 음성을 이용한 윈도우 제어 시스템은 응용 프로그램 호출 명령 5개, 키보드 제어 명령 22개를 실행시킨다. 그리고 제스처를 이용한 윈도우 제어 시스템은 호출명령 5개, 키보드 제어 명령 5개를 실행한다. 제안된 시스템에서 키보드 제어 및 응용 프로그램 호출 실행 명령은 음성 입력이 효율적이었다. 마우스를 제어할 경우는 음성을 이용하여 마우스 위치제어는 매우 힘들기 때문에 제스처만을 이용하였다.

제안된 다중 모드 입력 시스템은 다양한 형태의 입력을 통해 복잡한 주변환경이나 사용자의 요구사항을 만족 시킬 수 있는 대안이 될 수 있다. 그리고 키보드와 마우스등을 이용하여 제어하기 힘든 차세대 컴퓨터인 헤드 마운트(head-mount) 컴퓨터와 같은 웨어러블(wearable) 컴퓨터 시스템의 제어 수단으로 다중 모드 입력 시스템을 사용할 수 있다. 하지만 차후로 키보드를 완전히 제어할 수 있는 시스템 연구가 필요하며, 사이버 글러브 및 폴리머스 센서 같은 장비의 도움없이 제스처를 입력받아 제어할 수 있는 시스템의 연구가 필요하다.

참고 문헌

[1] D. J. Sturman and D.zeltzer, "A Design Method for Whole-Hand Human-Computer Interaction", *ACM Trans. on Information Systems*, vol. 11, no. 3, pp. 219-238, July 1993.

[2] S. S. Fels and G. E. Hinton, "Glove-Talk II - A Neural-Network Interface which Maps Gestures to Parallel Formant Speech Synthesizer Controls", *IEEE Trans. on Neural Networks*, vol. 8, no. 5, pp.977-984, 1997.

[3] J. Yang, Y. Xu and C. S. Chen, "Human Action Learning via Hidden Markov Model", *IEEE Trans. on Systems, Man and Cybernetics* - Part A: System and Humans, vol. 27, no. 1, pp. 34-44, Jan. 1997.

[4] H. Hamansky, "Perceptual Linear Predictive(PLP) analysis of speech", J. Acout. Soc Am., 87(4), pp. 1738-1752, April 1990.

[5] 한 영원, 배 건성, " Windows 95 환경하에서의 음성 인터페이스 구현", 전자공학회 논문지 제 34권 S편 제 5호, pp. 86-93, 1997.

[6] J. Richter, *Advanced Windows*, Microsoft Press, 1995.

[7] A. Baker, *The Windows NT Device Driver Book*, Prentice Hall PTR, 1997.